The American University in Cairo
Computer Science Department
**CS 447 – Spring 1999**

**Midterm Exam** **Date** : April 26, 1999
**Professor** : Muhammed Mudawwar **Duration** : 75 min.

**1.** (25 pts) Consider the following grammar:

$S \rightarrow \mathbf{a}\ S$
$S \rightarrow S\ B$
$S \rightarrow \mathbf{d}$
$B \rightarrow B\ \mathbf{b}$
$B \rightarrow \mathbf{c}$

**a)** Calculate the FIRST and FOLLOW sets for all nonterminals.
**b)** Calculate the Predict sets for all productions. Is this grammar LL(1)?
**c)** Give evidence that this grammar is ambiguous by showing two parse trees for the same string.

**2.** (30 pts) Consider the following grammar G:

S → ( L )
S → **id**
L → L **;** S
L → S

**a)** Eliminate left recursion. The resulting non-left-recursive grammar G2 should be equivalent to the original grammar G.

**b)** Calculate the Predict sets for all productions of the non-left-recursive grammar G2.

**c)** Construct the LL(1) parsing table for G2.

**d)** Using the nonrecursive predictive parser, show the parsing of the following input string (**id ;** (**id ; id**) **;** ). At each step show the content of the stack, remaining input, and parser action.

**3.** **a)** (8 pts) Write an unambiguous context-free grammar for a nested statement blocks where the semicolon separates the statements. A block is surrounded by **begin** and **end** and should have at least one statement. The last statement in a block should NOT be followed by a semicolon. Here is an example:

**begin** statement **; begin** statement **;** statement **end ;** statement **end**

**b)** (8 pts) Write an unabiguous context-free grammar for $\{wa^n b^n w^R \mid n \geq 1\}$ where $w$ is a string of $a$'s and $b$'s and $w^R$ is the reverse of $w$.

**c)** (7 pts) Write a regular expression that matches all strings of 0's and 1's that do not contain the substring "110".

**4.** **a)** (10 pts) Show an NFA that will match the keywords: **if, in, int, else, end, read, real**, as well as any identifier that consists of one or more lowercase letters only.

**b)** (12 pts) Convert the NFA of part *a* into a DFA. You can obtain the DFA intuitively without applying the conversion algorithm.